Coding Assignment 3 : Connect-N

Add proper comments (10%)

Good comments and documentation separates a good program from a bad one. You should add proper docstring and comments that explain your code.

Part 1 - Text-Based Connect-N (50%, Due 30/5)

In part 1, you will implement a text-version of the Connect-N game. The required features include:

Introduction

In your final assignment, you will implement a **Connect-N** game. **Connect-N** is an extension of the popular **Connect-Four** game. Below we will first introduce you the **Connect-Four** game.

Connect-Four is a two-player connection board game in which the players first choose a color and then take turns dropping one colored disc from the top into a **seven-column, six-row** vertically suspended grid. The pieces fall straight down, occupying the lowest available space within the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of **four of one's own discs**. If you have never played the game Connect 4, you can find its rule and try the game here

https://www.mathsisfun.com/games/connect4.html

Note that you MUST build your Connect-N game based on the provided .py file.

You can add new functions, variables etc, but you have to implement the functions provided. Please read the comments inside the file carefully.

Connect-N Game

Connect-N game is a variation of **Connect-4** where the grid expands to **N+3 rows** and **N+2 Columns**. The winning condition becomes the first to form horizontal, diagonal line of **N of one's own discs**. Here **N** can be any positive integer number the user inputs, e.g. 3, 4, 5, 6... etc.

Below are the required features

- Board (10%)
  - you must use the board stored in the dictionary `game` (see the provided source code)
  - should handle different N
  - should be able to display itself using unicode
- Winning Condition (15%)
  - determine if a player has won the game
  - print out the steps both players take for the game
- Game Play (15%)
  - allow players to drop white and black disk in turn at the specific column
  - prevent users from committing illegal move (e.g. drop disk the the column is full)
- Move history (5%)
  - allow the display of every previous moves in the game
- Save / Load the game (5%)
  - allow the game to be saved and loaded

Part 2 - GUI (40%, Due 13/6)

Based on your code in Part 1, wrap it with a GUI and add additional features below

The required features are

- A design sketch about the GUI you would like to implement (5%)
- Players (5%)
  - provide a GUI that allows both players to input his/her names
  - player names should be shown on the GUI
- Visuals (20 %)
  - the connect-4 board
  - player names, move history
  - a dialog when a player wins the game
- Game Interaction (15%)
  - drop the disk by selecting the column (you do not need to show the animation)
  - a reset button that restart the game

Bonus - Simple AI (10%)

create a simple AI that plays with the player. For example, the AI can do something like

- randomly drop the disk (5%)
- always copy the component
- drop the disk at the column that result in the highest connected disk in either horizontal or vertical lines
- feel free to come up with your own rule. Please describe your rule in the document.